Towards automatic apparatus integration in Automation Remote Laboratories

Saher Arnous, Arnaud Lelevé, Khalid Kouiss, Patrick Prévot

Towards automatic apparatus integration in Automation Remote Laboratories

Saher ARNOUS* - Arnaud LELEVE* - Khalid KOUISS** - Patrick PREVOT*

* *LIESP, INSA Lyon, Université de Lyon, F69621 Villeurbanne Cedex*
saher073@gmail.com, arnaud.leleve@insa-lyon.fr, patrick.prevot@insa-lyon.fr

** *LIMOS, IFMA, Campus Scientifique des Cézeaux BP 265, F63170 Aubière*
Khalid.Kouiss@ifma.fr

RÉSUMÉ. *Dans le contexte du ELearning, les travaux pratiques à distance (téléTPs) sont devenus indispensables, comme en formation traditionnelle, ceci notamment dans les disciplines techniques et scientifiques. Le LIESP a débuté en 2002 une recherche visant à fournir une architecture ouvrant la voie aux échanges de scénarios pédagogiques de téléTPs lorsqu'ils correspondent à des dispositifs similaires (mêmes fonctions mais matériels différents). Simultanément, le LIMOS a développé un processus de conception automatisant la génération de code pour Automate Programmable Industriel afin de faciliter la conception et la génération de programmes de systèmes automatisés.. Cet article présente un projet fusionnant ces deux approches afin d'aider les concepteurs de téléTPs dans leurs tâches de conception et d'intégration de dispositifs de téléTP comportant des éléments d'automatisme.*

ABSTRACT. *In the context of E-Learning, remote hands-on training has become an insisting need as in traditional learning, especially in scientific and technical disciplines. Electronic Laboratories (ELabs) have been growing for the last few years. LIESP team started in 2002 a research aiming to provide a framework which helps towards exchanging ELab learning scenarios when they fit to similar apparatuses (same functions, maybe not the same hardware). Meanwhile, LIMOS team focused on a design process to automate PLC code generation to help to design and generate programs for industrial discrete systems. This paper presents a project of merging these approaches to help ELab designers to design and integrate apparatuses into ELab frameworks when these apparatuses are discrete systems.*

MOTS-CLÉS : *Téléformation, Travaux Pratiques, Télé TP, Laboratoires Virtuels, Laboratoires en ligne, Télé Laboratoires Interactifs, Système de Gestion de la Formation*

KEYWORDS: *E-Learning, Hands-on Training, Remote Laboratories, Virtual Laboratories, Online Laboratories, Remote Interactive Laboratories, Distant Laboratories, Learning Management System.*

Introduction

In a scientific education system, hands-on training goes side by side with theoretical learning which is only a part of the educational process but not all the process. It is a necessary part to deepen the students' theoretical knowledge and reduce the gap between what they learn in the classroom and what they will face in the real world. As well, virtual universities offer scientific education through the World Wide Web for a large number of students around the world. Such educative systems must offer practical training for their students to achieve a reliable level of education especially in scientific (e.g. mathematics, chemistry, physics, medicine) and technical (e.g. electronics, fluid mechanics, robotics) disciplines (Corter et al., 2007) (Ma et al., 2006). Actually, Electronic Laboratories (ELabs) became, as a part of a virtual university, an essential need to provide students with practical training.

In fact, classical educational systems present significant drawbacks concerning the hands-on training. Some researches, e.g. (Jain, 1997) (Langa, 2004), cited that the increasing number of students in scientific and technical colleges limits the capability to provide students with adequate practical knowledge. In addition, the obligation of presence in a certain place at a certain time may not be convenient for all the students. Moreover, up to date hardware equipments as subject of training are not always affordable in the university, not to forget the devoted time and technical staffs (Kikuchi et al. Kikuchi). Also, not all students live geographically in the same area as their colleges, which imposes financial constraints on their education. On the other hand, not all the universities can hold the expenses of obtaining all necessary hardware equipments for hands-on training, it is also expensive to maintain and repair them, if this is available (Langa, 2004). Finally, but not the last, professional laboratory technical staffs are not always available (Rapuano et al., 2006). Accordingly, distant learning can be understood either within a virtual university or even within a real one.

Thanks to the Information and Communication Technologies (ICTs), distant learning has progressively improved; we can find in almost every university a Learning Management System (LMS) to provide electronic resources for local and distant learning activities. ICTs are also used to provide learners with alternative solutions for hands-on training by means of simulation tools which are widely spread through specialized software (e.g. MatLab and Labview). But the real challenge is to enable users to access real hardware (rather than only simulators) through a web browser for the purpose of hands-on training, since the designers of Distant Laboratories (DLabs), or Remote Laboratories (RLabs) require good knowledge in industrial computing and automated systems besides their skills in didactics in ELearning systems. Not to forget the difficulties faced by managers of RLabs to provide the instructors with flexible methods to propose pedagogical activities on remote materials (an instructor of civil engineering may not have skills in ICTs to exploit a RLab platform). Nowadays, current ELabs are thought to be used beside LMS with learning scenarios (Kolberg et al., 2007), (Ozdogru et al., 2007), (Bagnasco et al., 2006). Scenario making was previously not supported through e-learning standard during the successive developments of ELabs where the efforts were focused on the means to teleoperate systems. The integration of RLabs in LMS has been slowed down according to the fact that first LMS were usual closed proprietary software systems that are often not customizable (Rapuano et al., 2006).

LIESP team began in 2002, according to this realization, a project focusing on a framework involved in an edition chain for ELab learning scenarios. We aimed at proposing effective common tools for the authors and instructors of ELab learning scenario whatever discipline or apparatus, namely enabling the creation of generic scenario, independent from the materials and adapting it on different platforms. We took into account existing E-Learning frameworks: authoring tools, Learning Content Management System (LCMS) and LMS. After the general observation of different types of ELabs we inspired from the Hybrid Laboratories (HLabs) a common solution to override the problem which prevented automatic adaptation of a generic scenario (on a given family of apparatuses).

Meanwhile, LIMOS has been working since 2004 on a Component-Based Approach for the design of discrete control to drive conveying systems. A methodology allowing to automatically generate the control programs has been proposed to provide an easy way to obtain source code compatible with the IEC 61131-3 standard. The methodology is based on a MDE (Model Driven Architecture) approach in which models are described using meta-models at each step of the process.

This paper introduces a collaboration between LIMOS and LIESP to help ELab designers to design their apparatuses. This project is founded on the notice that in a distant learning context, many operations have to be automated to permit the teleoperation of appliances which were manually handled. Therefore such automated systems have to be designed with two purposes: enabling their teleoperation and offering a network common interface to enable LMS cooperation in an Elearning context. Therefore, this project aims at providing a semi-automated design process to help designers in this work.

The rest of this paper is organized as follow: through the next section (1) we make general observation of the different forms of E-Laboratories. We present an overview of our model of ”generic” ELabs in section 2. Section 3 presents a general overview of the approach for system control design, and we introduce our collaboration project in 4. Last section closes this paper by a general conclusion.

1. E-Laboratories overview

Electronic laboratories (ELabs) are potentially related to distant learning context, but they are involved as well in the context of classical learning, especially in scientific and technical disciplines. In fact, one can find through the scientific literature several kinds of ELabs dedicated to answer specific needs in each scientific/technical discipline; for example: electronic, circuits and electricity (Baccigalupi et al., 2006) (Bellmunt et al., 2006) (Kikuchi et al. Kikuchi), mechanics (Ishutkina et al., 2004) (Ashish et al., 2006), Laser and light sciences (Montes et al., 2006), networking security (Keller et al., 2006).

The common point within the two forms of training is that ELabs are manipulated through computers, so we distinguish between local and distant ELabs. The literature also carries out some studies about the aspects of ELabs in addition to several comparisons between the different contexts and types of ELabs, (Baccigalupi et al., 2006) (Corter et al., 2004) (Lindsay et al., 2005) (Ashish et al., 2006) for instance.

Local Elabs: (Ma et al., 2006) (Corter et al., 2007) (Singer et al., 2005) (Sere et al., 1998)

As an attempt to overcome some of the drawbacks mentioned in section (1), computer was introduced in traditional hands-on laboratories. They offer software to help students to perform actions on a real system (preset appliances with a set of parameters, automate some actions, get, plot and record massive data, etc.). They also may offer simulation tools if there is no real system. Moreover, hybrid local ELabs merge computer software utilization on a real system beside simulation tools when the real system lacks some parts that might be used in the context of training or when the simulated part is hard (or dangerous) to be manipulated by learners. A comparison between ELabs and hands-on laboratories is carried out in (Corter et al., 2004).

Distant Elabs (DLabs) are, in general, real laboratories whose users control and use it remotely rather than in place. It should contain the same equipments as in local ELabs for experimentations in addition to new ones dedicated to enable their teleoperation through a LAN, a WAN (Kikuchi et al. Kikuchi) or the internet itself (Rohrig et al., 2003).

Remote Labs (Rlabs), also known as “Web-Based Experimentation” (Gillet et al., 2005), “Remote Interactive Laboratories” (Sivakumar et al., 2005), ... offer remote access to real laboratory equipment and instruments. They may also be called “Virtual Labs” but this naming is confusing. They have been proposed in many disciplines (electronics (Davoli et al., 2006), earthquake study for civil engineering (Kuester et al., 2007), automation (Bellmunt et al., 2006), robotics (Ramaswamy et al., 2006), etc.).

Virtual laboratories (Vlabs), (Torres et al. 2006) (Hugo et al., 2002) (Lallican et al.2007), rely on system simulation. This is a possible and widely adopted solution where no physical resources are available. This solution reduces the gap between theoretical and practical knowledge for the learners but it does not lead to real useful experience since such systems don't allow acting/reacting with physical components.

They are based on software simulation of real systems, and are applied for a variety of scientific fields (for example: electronics (Li et al., 2005), power engineering, sensor networks (Christou, 2007), etc.).

Hybrid Laboratories (HLabs) are DLabs which remotely handle real systems beside simulations. HLabs represent a more flexible solution than the two others (RLabs and VLabs) since it contains a software part (simulated part) that can be virtually handled rather than physically (which is usually more difficult).

Like a VLab, the simulation part here can play a role of preparing for training for a first time. It emulates a real performance of a real part of a device or an instrument (Emami et al., 2008) (Ashish et al., 2006). This case happens when the simulated part either does not exist (e.g. expensive piece or defused one) or when this part is hard to be connected to the system for the purpose of reading information about its performance (e.g. measuring the temperature of the magnetic core of an electric engine). Also simulation is efficiently exploited, like in (Torres et al. 2006), where the video observation might not be sufficient for the user; therefore a simulation interface is used to give 3D view of the corresponding operation robot arm.

A scientific literature study leads to notice that the interest of researchers, concerning the design of such laboratories, is concentrated on two major points:

1. The design for the purpose of an efficient and reliable connectivity with the outer side of an RLab (namely the users via a network or the internet). In this case we found an attractive indication and use of the concept of Portals as in (Emami et al., 2006) (Landi et al., 2006), and as proposed for a future improvement in (Hugo et al., 2002).

2. The modelling of the physical structure of laboratory constituents leads to a better realization of the material part of the system, which simplifies the design of software to control and exploit the material resources of the laboratory. This point is interestingly discussed and a considerable attention is made on those works about automated systems. In fact, we may find at this point different ideas about the modelling of such systems. Each one proposes its own point of view to prove the applicability of his/her model on an automated system, which leads us to the fact that the cases (problems) in this domain don't have one ideal solution but can be adapted

and handled form different points of view, (e.g. two different models and methodologies of almost two similar convoying systems were presented in (Chiron et al., 2005) (Lallican et al.2007)). However, modelling tools, like UML2.0 and SysML, were used or proposed for this aim which proved, according to the authors, their efficiency in this domain, which in turn encouraged us to go forward in an observation of these tools for the goal of using them in our research.

2. LIESP Generic Elab model overview

In this section we illustrate the general aspects of our previous works about Elabs. Hacen Benmohamed has proposed in his PhD (Benmohamed, 2007) work both an Elab architecture enabling their use through learning scenarios written within Elearning standards. He designed a lifecycle opening the way for reusing learning scenarios for similar Elabs systems (same functions but possibly not exactly the same hardware). This work defines how to reuse standard Elearning tools to create (authoring tool), diffuse (LMS) and exchange (LCMS) ELab learning scenarios between laboratory platforms.

Hence, Elab learning scenario authors can create, manage and distribute their scenarios over a LCMS as classical training contents. Instructors can also work on similar systems where they are able to use such scenarios. As well, authors and instructors can search scenarios compliant with their systems and download them from ELab scenario libraries (LCMS). They then may enhance them to satisfy new needs and then redistribute new versions of those scenarios.

2.1. Scenario and genericity towards apparatuses

The IMS Global Learning Consortium creates standards for the development and adoption of technologies that enable high-quality, accessible, and affordable learning experiences. The IMS Learning Design specification supports the use of a wide range of pedagogies in online learning. Rather than attempting to capture the specifics of many pedagogies, it does this by providing a generic and flexible language. This language is designed to enable many different pedagogies to be expressed. The approach has the advantage over alternatives in that only one set of learning design and runtime tools then need to be implemented in order to support the desired wide range of pedagogies.

IMS-LD defines the structure of a learning scenario: a scenario written by an author (generally a teacher), delivered by an instructor towards learners. The standard defines resources (documents and web), activities for each user and how they may interact with each others during the corresponding training session. LD has been designed to be flexible so as it does not embed any didactic approach: scenario author is responsible for this didactic approach.

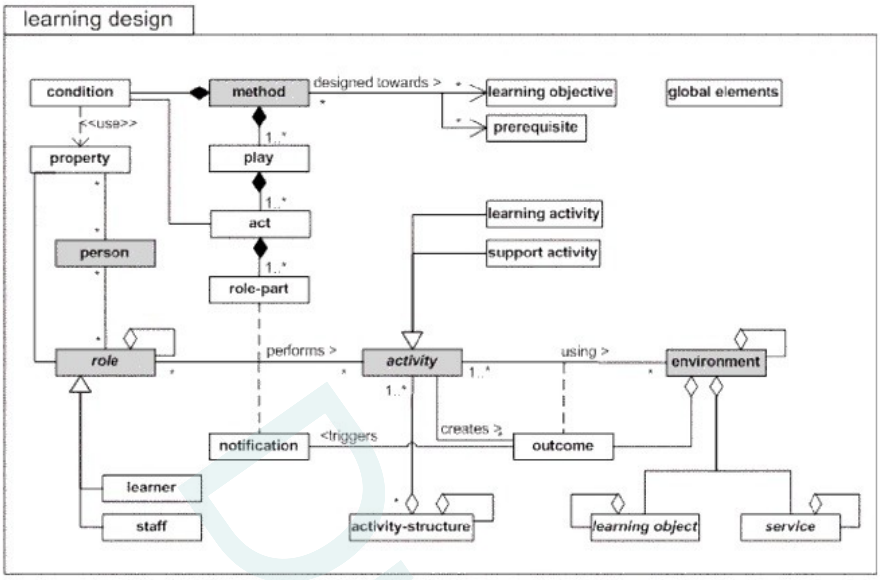


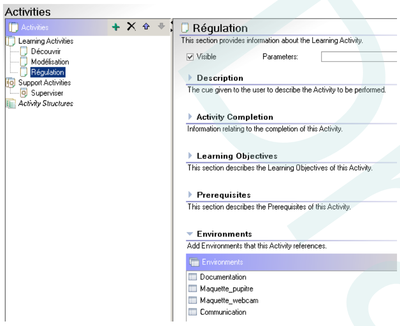
Figure 1. *IMS-LD conceptual model*

A learning package is composed of specifications about pre-requisites and the global learning objectives, in addition to temporal organization of the activities of different roles (Fig.1). It is also possible, according to IMS-LD, to define several simultaneous plays, each of which consists of sequentially running actions. Therefore, the training scenario can be applied on different learning groups of different chronologies.

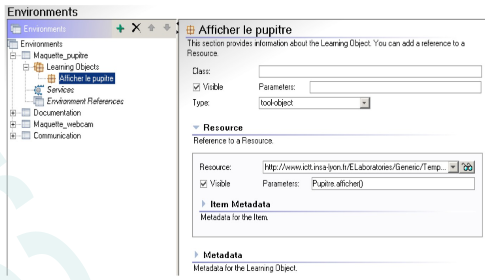
In turn, each action is composed of activities, each one is associated to one or more roles (learner, tutor… according to design needs). Every activity has a description, learning objectives, pre-requirements, ending conditions and one or more environments. Each environment is a collection of learning objects (like documentation) and necessary services for learners and instructors. The classical steps of a remote training scenario correspond to activities for the learners and the instructors (Fig.2). These activities are presented within environments that are responsible of distributing the pedagogical resources according to appropriate needs for each activity and each role.

According to IMS-LD standard, tools must be declared in the environments of a scenario as references (URL). When such scenarios are exchanged and used in different institutes, such links have to be manually replaced by local equivalent ones. In the context of Elabs, authors have to declare manipulation tools (teleoperation panel, video, plots, …) to permit users to use the corresponding apparatus. The same problem arises when the scenario is linked to another equivalent apparatus: every URL has to be manually updated (local apparatus address, its HMI address, portal address… this is impractical). To make sure that the scenario is generic, these URLs should describe the referred tool and its expected functions rather than a static address of a given manipulation server to avoid restricting the scenario to a specific one only apparatus in the world.

Thus, to use a generic scenario on equivalent apparatuses it is important to create type models of apparatuses that can satisfy clearly identified pedagogic needs. We call a type model "Template" in analogy with the techniques of object programming, where a template is a class which defines generic functions (common ones in every apparatus of a given family) that can be adapted according to programmers needs.



Environment definition used for the activities with IMS-LD "Reload" editor



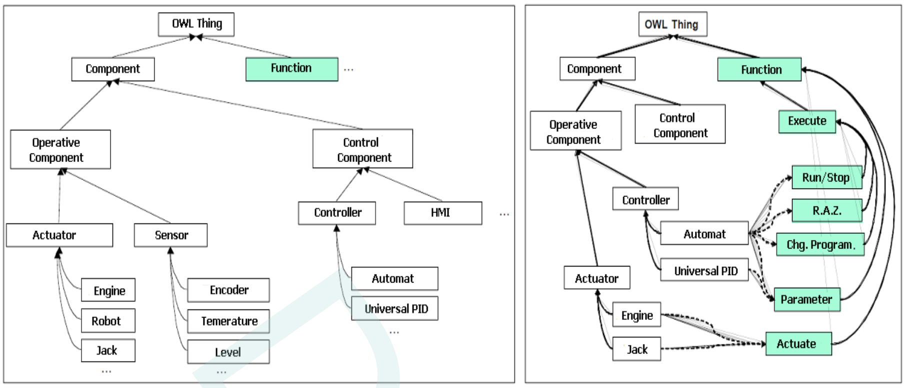
Environment definitions in a scenario

Figure 2. *Environment definition for a scenario*

2.2. Formal description ontology:

Templates management should be semi automatic to help scenario and apparatus designers. Therefore, both of software and humans should be able to manage the semantics of a template and other related representations; so we used formal description tools leading to a specific vocabulary represented under a form of ontology. Web semantics techniques are inspired and the standard OWL, normalized by W3C, is adopted.

Core ontology, called "Root Ontology", is constructed to describe the classical components of a pedagogic apparatuses associated to the functions they provide. This distributed ontology, dealt by a web server, has to be unique to provide reference to the ensemble of the architecture, but it is not exhaustive and has to be associated to extension tools. (Fig.3-a) shows some components of the root ontology where the arrows represent inheritance links, while (fig.3-b) illustrates the association between components and functions; dashed arrows correspond to an association of form: "*component X provides function Y*".



a) components b) component-function association

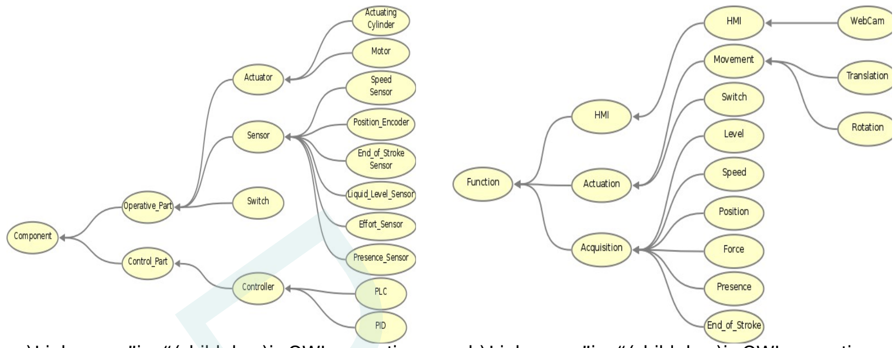
Figure 3. *Extract from Root Ontology*

Root ontology can be considered as an abstract description of apparatus parts one can find in any ELab (defining classes: Component, Function, Operative Part Component, Actuator, Sensor, ...). The first version of this system proposes a unique and extensible root ontology. New Components and Functions are built by following the same inheritance rules as in object programming to provide concrete part definitions (Temperature Sensor, Jack, PID Controller...). Next, child ontologies group some of these components are related functions to describe a familly of apparatuses (inverted pendulum, spectrometer, …). These ontologies are "familly templates" to help to classify each apparatus present in Elabs in order to associate to them corresponding learning scenarios. At the end of the chain, there are two other kinds of templates, related to a unique family template:

- apparatus templates which declare effective components and functions present in a given apparatus in a real ELab;
- scenario templates which define functions required for a given scenario.

2.3. Ontology illustration:

When we want to add a new family of apparatus in ELabs, a new template for this family must be written. (Fig.4-a) represents a set of component subclasses which can be found in an apparatus and (fig.4-b) shows a few usual functions which may be used for any template requiring them. As for components, this hierarchy is extensible to provide a larger set of functions



a) Links mean "is a" (child class) in OWL semantics   b) Links mean "is a" (child class) in OWL semantics

Figure 4. *Hierarchy of root classes for apparatus description*

As one can previously see in (fig.3-b), we defined links between components and functions in a bidirectional way in the following sense: component_provides_function and function_provided_by_component. This definition helps designers to associate functions of their choice to certain parts of the apparatus, as for the designer, a part of the apparatus is represented by a component.

These root classes are defined in the root ontology (a single file which is provided online on a public web server), so that it is shared by any child ontology (templates).

After having defined the root ontology file, template ontologies can be created by inheriting from the root ontology any necessary component and function.

2.4. Installing an apparatus:

To install a new apparatus in an ELab, the administrator of the platform declares it in the Electronic Laboratory Management System (ELaMS) managing the platform of the laboratory. He has to assign to this apparatus its apparatus template describing the functions really provided by it and the way to access them (from the list of the available ontologies of a given family).

At this stage, each class member is tagged with an OWL annotation which gives the URL to call to run this function. The ontology can also declare components corresponding to parts of the apparatus and defines which component provides which function.

(Fig.5) illustrate the process of installing an industrial oven. This process aims at installing a pedagogical apparatus to enable learners to regulate the temperature of an oven, through three levels of ontologies: starting from the root ontology by selecting a graduator component, then assigning a template of generic oven with the root ontology and associating the functions it provides by the chosen component(s) and finally instantiating the corresponding classes to the parts of the oven and their provided functions.

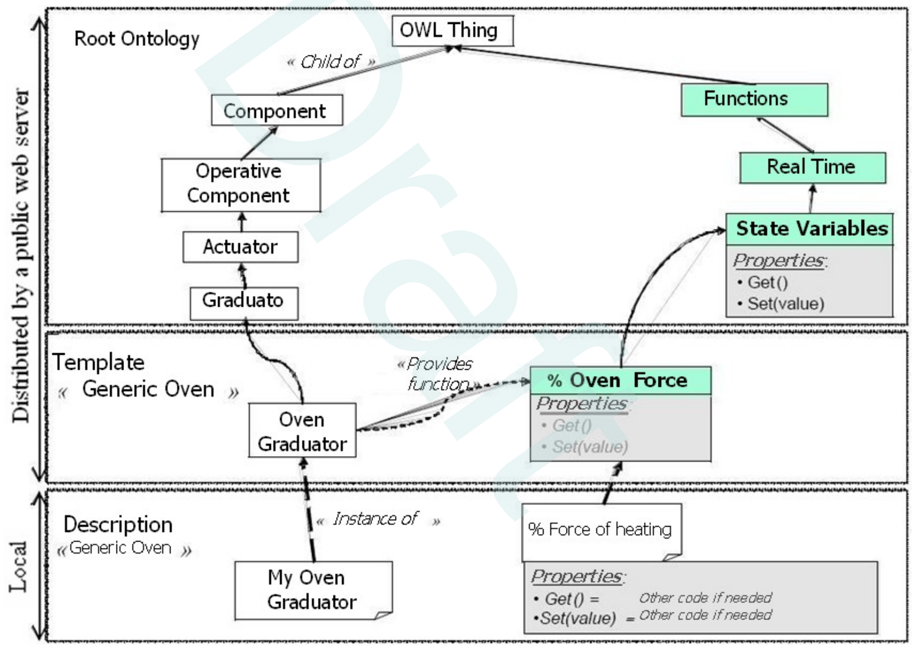


Figure 5. *Links between the 3 levels of ontologies for a Graduator class and its state variables*

2.5. Scenario edition and using phases:

When an author writes a generic scenario of an apparatus of a given family (e.g. industrial oven), he assigns to it an ontology which describes the functions used in this scenario; and for every used function in the scenario an instance from the apparatus ontology is added. Actually, the author has to make links between the scenario and the components and the functions in a manual way, and then the scenario is stored on a public LCMS server.

When an instructor searches on a LCMS for a scenario of his preferences, this research might result in a scenario associated to a same template as the apparatus or, if not exactly available, associated to similar template. Then he integrates it in the platform by addressing to the ELaMS (which manages the platform) a compatibility verification process. By means of dedicated software, ELaMS analyzes the required functions in the scenario if they correspond to the instances of functions from the apparatus template.

If every required function is instantiated in the apparatus ontology, the scenario is compliant with the apparatus, and then the ELaMS automatically adapts the scenario to the designated apparatus thanks to the already elaborated links template↔apparatus. The resources and the parameters of pedagogical objects of the scenario are transformed into URLs referring to the web interface of the ELaMS server, which intermediates with the apparatus corresponding to the requirements of the scenario. Like that, the scenario passes from generic phase to specific phase and becomes ready to be executed by LMS like any other classical pedagogical contents. For instance, when an instructor calls a pedagogical object, its web browser consults the ELaMS web server which in turn analyzes the request to deliver it to a free apparatus. ELaMs is charged at this stage to decide which free apparatus to reserve for the request session sent by the user.

In case that certain functions are not provided by the apparatus, due to the verification process, ELaMs notifies the instructor for the detected lack and the corresponding learning activities that might be affected, then the instructor can proceed in two ways: firstly, he can semi-automatically adapt the generic scenario, by manually defining the links between the scenario and the apparatus, if he estimates that the apparatus can provide the undetected function by ELaMS. Secondly, when the instructor finds that the apparatus cannot provide certain functions, he may enhance the apparatus or rewrite the corresponding activity to override this situation.

To avoid the manual intervention of the instructor, when certain instances in the scenario ontology are not tagged in the apparatus template, it is possible to use software to simulate the lacked functions in the apparatus and provide them as if they already exist. Of course, the software is tagged through URLs which target the simulator functions. This way it will be very easy to replace the simulator functions with real functions when upgrading the apparatus to a newer version (or even replacing it by a newer one) so that it provides the lacked functions. Moreover, the scenario author can get advantage of the simulator to integrate ELabs resources in his scenario, no matter if he cannot reach a real apparatus. In fact, writing a scenario this way regardless of a specific apparatus guarantees the genericity of the scenario, where there is no need to adapt it to a specific apparatus for the purpose of test.

2.6. Apparatus virtualization interface

In this paper we are trying to provide ELabs with a flexible interface to enable automatic (or semi-automatic) scenario adaptation in the platform, so that it can be executed on a pedagogic apparatus (of a given family) without counting on the material setup of the apparatus.

In fact, almost all the apparatuses within an ELab are equipped with electronic control interface (control panel) to enable their connectivity with computerized systems. This interface is responsible of receiving electronic commands from a user, analyzing them and redirecting the command to the designated part (component) of the apparatus, with respect to the apparatus setup (voltage, current force, heat, etc.). Also it receives feedback signals from the sensors attached to the parts of the apparatus (if they are available), and transfers them to the user with respect to the computerized system. Such an interface is controlled in an embedded way through local software (installed in a chip on it), or through dedicated hardware which can receive and understand commands from specific software on the user side. According to this realization, when we associate an apparatus to ontology, and declare components corresponding to parts of it, and define links between components and the functions they provide, the commands sent to actuate a certain function of a certain component pass through the electronic interface of the apparatus which guarantees the regular performance. This is not a problem for an activity in a specific scenario of a specific apparatus. But in our project we are trying enable a generic scenario adaptation on an apparatus of an unknown setup. For example, if an activity in a generic scenario imposes to rotate an electric engine, where there might exist two types of engines: small one (12 volts) and large one (220 volt), this means that the scenario must specify which component should be used. Simply this means that the scenario loses its genericity.

To overcome this problem we proposed to associate the apparatus with a virtual interface which describes only its functions regardless of its physical setup. For the virtualization mechanism, it is based only on functions; therefore it is unnecessary to define components for its functioning. Instead, components definition and their associations with some functions can be used for the template ontology. Like that, a family of apparatuses will have the same functions, while some components handling these functions may be different from one apparatus to another within the same family.

3. Component-Based approach for System Control Design:

LIMOS team began in 2004 a project, through which Fabien CHIRON proposed in his PHD work (Chiron, 2007) a component-based approach for the design of discrete control to drive conveying systems. A methodology allowing to automatically generate the control programs has been proposed to provide an easy way to obtain source code compatible with the IEC 61131-3 standard. The methodology is based on a MDE (Model Driven Architecture) approach in which models are described using meta-models at each step of the process.

Founded on the concept of Component Based Software Engineering, the approach aims likewise at reducing the required time to design the control part, in the context of conveying systems, and to facilitate the creation of controls in the context of reconfiguration, since it is necessary to provide several versions of control. In this case the goal is to facilitate the creation of these controls.

3.1. Design process

The global process is part of a usual flow based on a simulation to validate or modify the design parameters. The objective of this process is to design, validate and implement control of conveying systems. (Fig.6) describes the steps involved by the design process:

- system modelling,
- generation of material part model,
- generation of control programs and
- simulation.

The system model is built upon a component library. After validation, control programs can be loaded in PLC(s) (Programmable Logic Controller). If simulation results do not correspond to the specifications, the system model is modified.

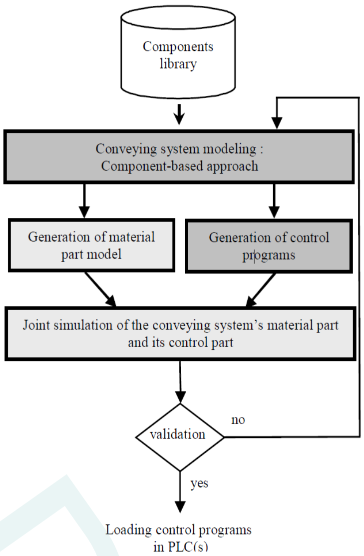


Figure 6. *Global design process*

3.2. Component-Based Approach:

This approach provides a clear and easy way to reuse previously modelled elements or to modify the system's internal structure. If the study is based on an existing system, the first step consists in a structural splitting up in order to get components.

Components refer to operations which are performed by a resource of the system. The resources in turn (ex: sensor, hydraulic jack) can perform several operations which implement the resource functionalities.

Based on the typology applied to generic and contextual functions, three different types of operations are applied:

1) Basic operations like advancing by a jack or detecting by a sensor.

2) Contextual operations like detection the position of a jack by a sensor where the sensor is associated with the jack.

3) And effective contextual operations like transferring an object from one area to another by a jack on a conveyor (Illustrated in (Lallican et al.2007)). The typology of operations is represented by a class diagram (Fig.7).

A component is a set of operations including monitoring, supervision and control points of view. Besides functions, it takes into account the system structure and its physical organization. Components types are defined by analogy to operations types, also detailed in (Lallican et al.2007):

1) Basic component which is a set of basic operations e.g. jack or sensor.

2) Basic enriched component which is a set of operations enriched with contextual ones, like a jack to which a sensor is attached (sensor here provides its own operations in addition to those of the jack).

3) Support component whose only function is to support and specify some spatial constraints, e.g. a belt conveyor is viewed as a support component; it enables to define an area of admissible evolutions for parts (parcels, products), this area can be straight or curved for a conveyor.

4) Effective contextual component is a set of effective contextual operations put together, according with the part flow, e.g. a jack component and a motor component associated with a conveyor component enable to define an ejector component. The ejector component has two operations: transfer from area 1 to area 2 by motorized conveyor, and transfer from area 1 to area 3 by the jack and conveyor.

5) System component which models the whole system and refers to at least one effective contextual component.

The component description uses a black-box formalism. Inputs and Outputs relating to physical flow are separated from Inputs and Outputs dedicated to control (Fig.8). Basic and support components include parameters providing adaptability to different designs. They are stored in a library as validated ready-to-use models. An aggregation procedure has been developed. It consists in building a component of level L from several components of level L-1 brought together. Contextual components represent the first level of aggregation while the system component is the last level of aggregation (the whole system).

As explained in (Lallican et al.2007), a component is composed of four views: operating part, graphical, constraints and control views. Control part is described using sequential function charts (SFC). SFC has the advantage of manipulating simple concepts which are comely used by PLC program developers. When components of level L are selected to be aggregated, the co-ordination of the different control parts, named hierarchical control part, has to be generated for the level L+1 component.

The control structure is hierarchical, and two kinds of control part are considered: low level control part and hierarchical control part. Basic and support components which are stored in a library include low level control part. A hierarchical control part refers to an aggregated component.

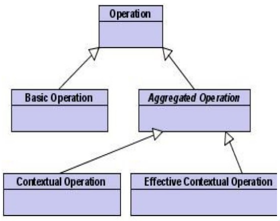


Figure 7. *Typology of operations*

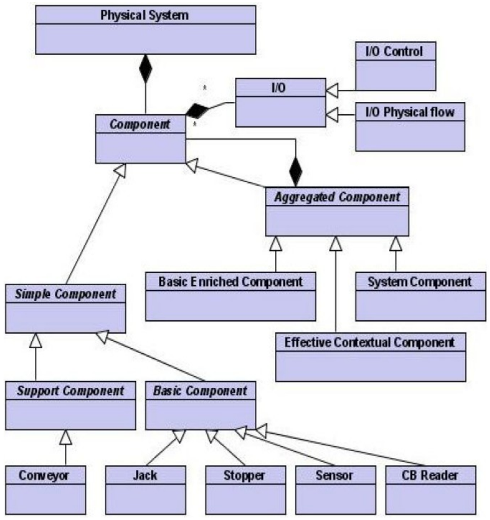


Figure 8. T*ypology of components*

3.3. Control Design

The control design methodology (Fig.9) involves three steps:

- conveying system modelling,
- generation of hierarchical control views and
- generation and partitioning of control programs.

The first step is dedicated to create the partial component model and the control system model by using a components library. A partial component model is seen as an assembling of components. It is partial because it does not contain control views of aggregated components. An algorithm (details in (Lallican et al.2007)) is proposed to automatically generate the control views of aggregated components. This algorithm is divided into three successive phases (A, B and C) uses a library of control templates. Phases A and B allow to generate the control views of both basic enriched components and effective contextual components, while phase C presents a new function called "constraint view simplifications" which expresses the conditions for beginning and ending the activation of effective contextual operations. Constraint views are related to effective contextual components and to the system component. This function allows simplifying the different constraints to have only one constraint by effective contextual operation. Thus, the partial component model is the reference from which the hierarchical control part generation step is performed to obtain a whole component model. In turn, component model is refined to obtain the whole component model.

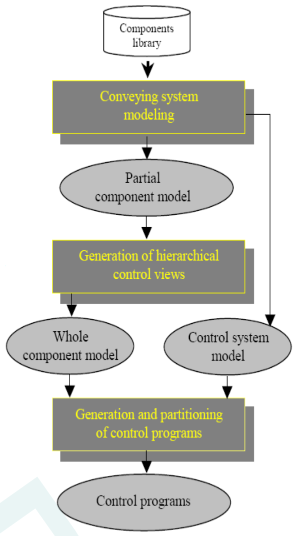


Figure 9. *Control design methodology*

Both the control system model and the whole component model are used in the step of generation and partitioning of control programs, to generate control programs. The control programs generated are IEC 61131-3 compliant, implemented by PLCs without any transcription and are expressed using XML.

In this step the control system model captures all aspects of a control system in terms of implementation (hardware components) and the whole component model is used for the description of control functionalities.

The approach used here follows the Model Driven Architecture methodology, proposed by the OMG. The system functionalities are defined by the Platform Independent Model (PIM), to which the component model corresponds. The projection of functionalities on the hardware architecture defines the Platform Specific Model (PSM). Thus in this approach, the PSM corresponds to control programs which can be implemented on PLCs. The hardware architecture which is mainly composed of PLCs, is described in the control system model.

4. Synthesizing of future development points

Previously presented works are a basis for a collaboration between LIMOS and LIESP to help ELab designers to design their apparatuses. This project is founded on the notice that in a distant learning context, many operations have to be automated to permit the teleoperation of appliances which were manually handled. Therefore such automated systems have to be designed with two purposes:

- enabling their teleoperation and

- offering a network common interface to enable LMS cooperation in a Elearning context.

Therefore, this project aims at providing a semi-automated design process to help designers in this work, based on previous works of LIESP and LIMOS teams.

Currently, Elab designers have to design :

- the operative part (sensors, actuators) to enable automatic handling of an initially local apparatus;
- the control part embodied by a PLC or a PC;
- the network interface to enable teleoperation and LMS synchronization. This interface is based on the familly template definitions which describe which apparatus function should be available remotely.

Control part and network interface design are really dependant : through the network interface, users will be able to run control part functions. So the design of both ones should be linked to prevent a double design and to prevent errors. When starting from scratch, the requirements for an Elab are defined from learning needs. These learning needs can then (manually) be converted to Elab components and required functions which can be embodied by a part of or a whole familly template.

The main idea is to use the LIMOS edition chain to automate the source code generation from the corresponding familly template definitions for:

- the control part (PLC program skeleton) : desired functions and how to call them from a URL;
- the interface : filling the apparatus template with correct URL
- (in a second step) the web HMI: reusing widgets from a library, with corresponding parameters automatically set.

We remarked in the current stage of the project that there are several points to be improved.

The root ontology we presented is in its beginnings and requires regular editions. The fact that it is stored in a single file and available online means risking modifying some definitions already used by child classes. At this time we may propose to omit modifications and replace it with the possibility of overloading undesired descriptions (which are desired to be modified) by new ones, with conformance to overloading concept used in Object Oriented Programming (OOP).

At verification process of the compatibility (template ↔ apparatus), when ELaMS notifies the instructor of the lack of some function, he may manually define links between the scenario and the apparatus. This means that the formal description in the ontology was insufficient to represent all the available functions of an apparatus. In this case we can propose a method to automatically generate a sub ontology from the ontology associated to the apparatus, by inheriting its functions and adding to it the manually defined functions by the instructor. This way, when another scenario linked to the same ontology passes though the verification process, it can have the possibility of being automatically adapted. In parallel, simulation software can be employed to provide unavailable functions of an apparatus so that the generic scenario can find corresponding functions to its activities.

Inspired from the automated production systems, another idea is to propose an automatically executable generic scenario within ELaMS platform as a utility, to help learners to understand the essential functions of an apparatus (of a given family). Like an illustrative video clip, such a scenario would help the learners to observe real functioning of the apparatus (with the possibility to repeat some activities), in addition to read and interact with results (feedback). This can be considered as a preparation for the learner to be familiar with the environment he/she is about to use. Modifying the sequence of certain activities in such a scenario leads to new results, but it uses always the same functions of the apparatus. This idea could also be creatively used in automated production systems (with respect to another platform than ELaMS) for staff training. Its automated generation would simplify the operations of software update and afford a time gain in creating new training scenarios by reusing parts from a library.

Conclusion

Electronic laboratories have become an essential need in the core of distant learning as well as in classical learning. However, they are not fully integrated in Learning Environments yet. As apparatuses are often made of different hardware even if they fill the same learning purposes, this situation slows down the dynamics of exchanging and reusing ELab learning scenarios. Authors from LIESP and LIMOS laboratories have began to work on a semi-automated design process to help ELab designers to provide apparatuses based on automated discrete systems which enable LMS synchronization and generic learning scenario reusability. This paper sums up previous works of both teams and depicts basic lines of this common project.. Automating the design process of the control part, the network interface and the HMI should permit to get time and cost gains. The concept could also be extended to automatic generation of demo scenarios and staff training on specific industrial automated systems.

References

Ashish Mani, Patvardhan C., "*A study of ict enabled laboratories*". In Annual India Conference, 2006.

Baccigalupi A., De Capua C., Liccardo A., "Overview on development of remote teaching laboratories: from labview to web services". *In Instrumentation and Measurement Technology Conference, IMTC 2006. Proceedings of the IEEE*, pages 992–997, 2006.

Bagnasco A., Buschiazzo P., Ponta D., Scapolla M., "A learning resources centre for simulation and remote experimentation in electronics," *In Proceedings of the 1st international conference on Pervasive Technologies Related to Assistive Environments*, New York, USA, ACM, pp. 1–7, 2008,.

Bellmunt O.G., Miracle D.M., Arellano S.G., Sumper A., Andreu A.S., "A distance plc programming course employing a remote laboratory based on a flexible manufacturing cell". *Education, IEEE Transactions on*, 49(2):278–284, 2006.

Benmohamed H., ICTT@Lab: un environment pour informatique la génération et l'exécution de scénarios de téléTPs, Thèse de doctorat, L'Institut National des Sciences Appliquées de Lyon, 2007.

Chiron F., Kouiss K., *Distributed control systems: from design to real implementation with uml 2.0.*. Int Conf on Industrial Engineering and Syst. Management IESM 05, Marrakech, Maroc, 2005.

Chiron F., Contribution à la flexibilité et à la rapidité de conception des systèmes automatisés avec l'utilisation d'UML, Thèse de doctorat, L'université Blaise Pascal de Clermont- Ferrand, 2007.

Christou I. T., Efremidis S., Tiropanis T., Kalis A., "Grid-based virtual laboratory experiments for a graduate course on sensor networks", *IEEE Transactions on Education*, vol. 50, no. 1, pp. 17–26, 2007.

Corter J.E., Nickerson J.V., Esche S.K., Chassapis C., "Remote versus hands-on labs: a comparative study", *In Frontiers in Education, FIE. 34th Annual*, pages F1G–17–21 Vol. 2, 2004.

Corter J. E., Nickerson J. V., Esche S. K., Chassapis C., Im S., Ma J., "Constructing reality: A study of remote, hands-on, and simulated laboratories," *ACM Trans. Comput.-Hum. Interact.*, vol. 14, no. 2, p. 7, 2007.

Davoli F., Spanospano G., Vignola S., Zappatore S., "Labnet: Towards remote laboratories with unified access", *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 5, pp. 1551–1558, 2006.

Emami M.R., Helander M., *edesign*. Proceedings of the Second CDEN International Conference on Education, Innovation and Practice in Engineering Design Kananaskis, Alberta, Canada, 2006.

Emami M.R., Martin A., *Dynamic load emulation for robotic hardware-in-the-loop simulation platforms*, IEEE International Symposium on Industrial Electronics Cambridge, UK, 2008.

Gillet D., Ngoc A. V. N., Rekik Y., "Collaborative web-based experimentation in flexible engineering education", *IEEE Transactions on Education*, vol. 48, no. 4, pp. 696–704, 2005.

Hugo L.G., Adolfo B., Guimarães A.S. Souza, *Internet based remote laboratory: The level control of three coupled water reservoirs*, IFAC, 2002.

Ishutkina M., Feron E., Casini M., Vicino A., "An internet based laboratory for control of a safety critical system", *In Systems, Man and Cybernetics, IEEE International Conference on*, pages 2707–2712 vol.3, 2004.

Jain R., *A revolution in education*, Multimedia IEEE, 1997.

Keller J., Naues R., "A collaborative virtual computer security lab". *In e-Science and Grid Computing*, Second IEEE International Conference on, pages 126–126, 2006.

Kikuchi T., Fukuda S., Fukuzaki A., Nagaoka K., Tanaka K., Kenjo T., Harris D.A., "Dvts-based remote laboratory across the pacific over the gigabit network". *Education, IEEE Transactions on*, 26–32, 2004.

Kolberg, S. Courivaud, D. Ozbek, M.E., *Lms and interactivity - technical issues for remote laboratories*, in Proceedings of IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2007), IEEE, Ed., Athens, Greece, Sept. 2007.

Kuester F., Hutchinson T.C., "A virtualized laboratory for earthquake engineering education", *Computer Applications in Engineering Education*, vol. 15, no. 1, pp. 15–29, 2007.

Lallican J.L., Berruet P., Rossi A., Philippe J.L., "A component-based approach for conveying systems control design", Kananaskis, Alberta, Canada, *In ICINCO-SPSMC*, pages 329–338, 2007.

Langa H.M.. "Design and implementation of a laboratory practicals management tool", *7th AFRICON Conference in Africa*, Vol.2, 2004, p. 965–968.

Landi C., Liccardo A., Polese N., "Remote laboratory activities to support experimental session for undergraduate measurements courses". *In Instrumentation and Measurement Technology Conference, IMTC 2006. Proceedings of the IEEE*, 2006, p. 851–856.

Li S., Khan A., “Applying it tools to a laboratory course for measurement, analysis, and design of electric and electronic circuits,” *IEEE Transactions on Education*, vol. 48, no. 3, 2005, p. 520–530.

Lindsay E.D., Good M.C., “Effects of laboratory access modes upon learning outcomes”, *Education, IEEE Transactions on*, 2005, p. 619–631.

Ma J., Nickerson J. V., “Hands-on, simulated, and remote laboratories: A comparative literature review”, *ACM Comput. Surv.*, vol. 38, no. 3, 2006, p. 7.

Montes M., Swiatczak J., Lesniewski P.J., John C.T., Golczak M., *Universal laboratory networking system (ulabnet)*”. In Instrumentation and Measurement Technology Conference, IMTC. Proceedings of the IEEE, 2006.

Ozdogru B., Cagiltay N., *How content management problem of a remote laboratory system can be handled by integrating an open source learning management system? problems and solutions*, in Proceedings of IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2007), IEEE, Ed., Athens, Greece, Sept. 2007.

Ramaswamy B., Chen Y., Moore K., *Omni-directional robotic wheel - a mobile real-time control systems laboratory*, In American Control Conference, 2006.

Rapuano S., Zoino F., “A learning management system including laboratory experiments on measurement instrumentation”, *Instrumentation and Measurement, IEEE Transactions on*, 2006, p. 1757–1766.

Rohrig C., Bischoff A., “Web-based environment for collaborative remote experimentation”. *In Decision and Control, 2003. Proceedings, 42nd IEEE Conference on*, Vol.3, 2003, p. 2514–2518.

Sere M.G., Leach J., Niedderer H., Psillos D., Tiberghien A., Vicentini M., Improving science education: Issues and research on innovative empirical and computer-based approaches to labwork in Europe, Final report of project labwork in science education, European Commission, Tech. Rep., 1998.

Singer S. R., Hilton M. L., Schweingruber H. A., “Americas lab report: Investigations in high school science, committee on high school science laboratories: Role and vision”, National Research Council, Tech. Rep., 2005.

Sivakumar S., Robertson W.,. Artimy M, Aslam N., “A webbased remote interactive laboratory for internetworking education”, *IEEE Transactions on Education*, vol. 48, no. 4, 2005, p. 586–598.

Torres Medina F., Candelas Herías F. A., Jara Bravo C. A.. *Flexible virtual and remote laboratory for teaching robotic*”, 4th International Conference on Multimedia and Information & Communication Technologies in Education. Sevilla. 200.

Annex

Vers la génération automatique d'interfaces de virtualisation de maquettes de téléTPs